Remarks

Claims 1-4 and 6-12 have been examined. The Examiner has rejected claims 1-4 and 6-12. By this amendment, claims 1-2 and 7-8 are amended. New claims 14-17 have been added. Thus, claims 1-4 and 6-17 remain in this application.

Applicant has reviewed the Examiner's rejections, as well as the newly cited art. Applicant appreciates the examiner's attention to the previous amendment. However, upon review of the Examiner's rejections, Applicant respectfully disagrees with the Examiner's conclusions.

Claim Rejections - 35 USC §103

The Examiner rejected claims 1-4 and 6-12 under 35 USC 103(a) as being unpatentable over Sharangpani (U.S. Patent No. 5,699,537) in view of Hirata (U.S. Patent No. 5,430,851). Applicant traverses.

Claim 1 is provided below, as amended, for ease of reference:

1. (Currently amended) A pipelined multistreaming processor, comprising:

an instruction cache for concurrently providing a plurality of instructions for a plurality of instruction streams;

fetch logic coupled to said instruction cache enabled to concurrently fetch said plurality of instructions for said plurality of instruction streams from said instruction cache;

a plurality of instruction queues coupled to said fetch logic where each one of said plurality of instruction queues is associated with at least one of said plurality of instruction streams, wherein the number of said plurality of instruction queues is greater than said plurality of instruction streams that are provided by said instruction cache;

a dispatch stage coupled to said plurality of instruction queues for selecting and dispatching instructions for said plurality of instruction streams to a set of execution units; and

select logic coupled to said instruction cache, and to said plurality of instruction queues, said select logic monitoring each of the plurality of instruction queues, said select logic selecting ones of said plurality of instruction streams to fetch instructions from said instruction cache, the selecting based on the monitoring.

The examiner stated that Sharangpani taught the invention as claimed including an instruction cache (206,304). Applicant notes that an instruction cache is not novel. However, what is claimed by applicant is an instruction cache that concurrently provides a plurality of instructions for a plurality of instruction streams. Sharangpani does not teach provision of a plurality of instruction streams. And, Hirata, although showing multiple ports from the instruction cache, is actually teaching the concurrent provision of multiple instructions, but from a single stream. (see col. 5, lines 18-22) repeated below:

> **instruction fetch unit by fetching three instructions from an instruction stream to deliver to an instruction setup unit in the first cycle and fetching three other instructions from another instruction stream to deliver to another instruction setup unit in the next cycle. Since**

That is, Hirata provides three instructions from a first stream in a first cycle. Then, three more instructions from a different stream in a second cycle. In contrast, Applicant's invention provides two or more instructions, for two or more streams, concurrently. Such is not taught by Sharangpani, taken alone or in combination with Hirata.

The examiner further stated that Sharangpani teaches Fetch logic (302, 402) to concurrently fetch instructions from the instruction cache. However, as originally claimed, Applicant's fetch logic concurrently fetches a plurality of instructions from the instruction cache, for a *plurality of instruction streams*. This is nowhere taught in either Sharangpani, or Hirata.

The examiner further stated that Sharangpani teaches a plurality of instruction queues (322, 415, 416, 417, 418). Applicant has examined Sharangpani and believes that the Examiner is referring to block 222 rather than 322. Applicant agrees with the Examiner that Sharangpani shows a plurality of instruction queues. However, the queues of

Sharangpani are all associated with a single instruction stream. Sharangpani's queues hold instructions that are dispatched, either in order (according to one embodiment), or out of order (according to an alternative embodiment). But, all of the instructions relate to a single stream. See Col. 6, lines 39-46 repeated below:

> as illustrated. The dispatch queues **415–417** receive instructions from the front-end **301** and dispatch the instructions to the corresponding execution cluster in the order in which the instructions were received from the front end **301**, in one embodiment. In an alternate embodiment, the dispatch queues **415**, **416**, and **418** are in-order dispatch queues, while the load/store dispatch queue **417** provides for out-of-order dispatch of some load operations. The operation of

The purpose of Sharangpani's queues is to attempt to keep each of the functional units, such as an integer execution cluster, occupied with instructions, for "efficient dynamic instruction scheduling and execution". See abstract. Applicant's invention, on the other hand, provides queues for a *plurality of instruction streams* for the purpose of decoupling the fetch stage from the dispatch stage, thereby reducing the number of ports required on the instruction cache. The categorization of instructions within Applicant's queues is based on instruction streams, NOT on which functional unit will execute the instructions. This is an important distinction that is nowhere taught by Sharangpani, taken alone or in combination with Hirata. Nothing in Sharangpani or Hirata teach associating instruction queues with instruction streams, much less having more instruction queues than the number of instruction streams that are provided by the instruction cache.

The examiner admits that Sharangpani did not expressly detail that the instructions that were fetched concurrently were part of a plurality of threads. However, the examiner stated that Hirata taught the concurrent fetching of instructions for a plurality of threads. Applicant respectfully disagrees. As quoted above, Hirata taught concurrent fetching of three instructions for a first thread, then three instructions for a second thread. Hirata did not teach concurrent fetching of a plurality of instructions for a plurality of threads.

Further, neither Sharangpani, taken alone or in combination with Hirata, teaches select logic which monitors the instruction queues (associated with instruction streams) which

selects ones of the instruction streams for fetching from the instruction cache based on the monitoring.

Thus, claim 1 taken as a whole, claims a processor that has an instruction cache that concurrently provides two or more instructions, for two or more instruction streams, two a plurality of instruction queues, the number of instruction queues exceeding the number of fetched instruction streams, wherein the selection of which instruction streams are fetched from the instruction cache is driven by selection logic which monitors the contents of the instruction queues. Such invention allows instruction queues to be filled for a number of instruction streams, while minimizing the number of ports required on an instruction cache.

Respectfully, applicant suggests that this combination, as recited in claims 1 and 7 (as amended) is nowhere taught, suggested, or even hinted at in Sharangpani, taken alone or in combination with Hirata.

For all of the above reasons, applicant respectfully requests the examiner to withdraw his rejection of claims 1 and 7.

All other claims depend either directly or indirectly from claim 1, and add further limitations that are neither anticipated nor obviated by Sharangpani, taken alone or in combination with Hirata. Applicant therefore respectfully requests that the examiner withdraw his rejection of these claims.

Applicant earnestly requests the examiner to telephone him at the direct dial number printed below if the examiner has any questions or suggestions concerning the application.

Respectfully submitted,

/JAMES W. HUFFMAN/

James W. Huffman
Huffman Law Group, P.C.
Registration No. 35,549
Customer No. 23669
1832 N. Cascade Ave.
Colorado Springs, CO 80907
719.475.7103
719.623.0141 fax
jim@huffmanlaw.net

Date: 1/19/06